

REMARKS

This is a response to the Office Action dated August 24, 2005. Claims 1-19 are pending. Claims 1-12, 16, 17, and 19 have been amended. Individual issues raised by the Examiner are addressed below in order in which they appear in the Office Action.

Objections to the Specification and the Abstract

In paragraph 3 of the Office Action, Applicants were requested to supply the serial numbers of the related applications cited on page 1 of the specification. The specification has been amended to comply with the request.

In paragraph 4 of the Office Action, the Abstract of the Disclosure was objected to because it exceeded the limit of 150 words. Applicants have amended the Abstract accordingly, and respectfully request that the new Abstract be entered in the file of this application and be used for all purposes in the future.

Claim Amendments

Claims 1-12, 16, 17 and 19 have been amended to more particularly point out and distinctly claim the subject matter, which applicants regard as their invention. The amendments were made solely to clarify the claim language and avoid potential ambiguities, and were not made in response to the prior art rejections, which are considered below.

Claim Rejections Under 35 U.S.C. §103(a)

In paragraphs 5-6 of the Office Action, claims 1-15 and 17-19 were rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,253,228 issued to Ferris et al. (“Ferris”), in view of U.S. Patent No. 6,167,523 issued to Strong (“Strong”).

In paragraph 7 of the Office Action, claim 16 was rejected under 35 U.S.C. 103(a) as being unpatentable over Ferris in view of Strong, and further in view of a publication by Microsoft Corporation, “Microsoft Component Services: Server Operating System A technology Overview.” (“Microsoft Technology Overview”).

Briefly, Ferris discloses a method and apparatus for updating and synchronizing information between a client and a server. (Ferris, col. 4, line 32 - col. 5, line 9). A user inputs data in elements of applets running in a browser on the client computer, and as a result, some data displayed to the user by those applets should be updated. An Applet Group Controller is a software module running on the client computer, which facilitates the transfer

of the input data from the applets to the server and the transfer of the data from the server to the client computer required to update information displayed in the applets. Several satellite software modules that facilitate the integration of the Applet Group Controller with the applets running on the client computer and with the server are disclosed. (*Id.*). According to Ferris, there are two benefits achieved by this arrangement. First, there is constant synchronization between the client and the server, and second, the user actions are recognized instantaneously without the need to submit the input data to the server by the means of the FORM tag of HTML and then to send an entirely new page in response to this submit operation back to the client computer. (*Id.*).

Strong discloses a method of entering data in HTML forms on the client computer, transmitting the data to the server and validating the entered data on the server by comparing the data entered into certain fields to corresponding predefined formats stored on the server in a collection of formats called a registry. (Strong, Abstract).

The present application discloses a validation manager that is instantiated in response to processing of a markup language file (e.g. HTML document). The validation manager runs on the client computer and validates data that the user inputs in the input fields created in response to processing the markup language file (e.g. HTML forms) before sending the data to the server.

According to the Office Action, the combination of Ferris and Strong renders claims 1-15 and 17-19 obvious under 35 U.S.C 103(a). Applicants respectfully disagree.

Claim 1 requires, among others, (1) receiving text input to the GUI displayed as a result of processing of a markup language file comprising a description of a GUI, the description comprising a GUI element enabled to receive text input (e.g. HTML form), and (2) the application, which processes markup language files (e.g. browser) instantiating the validation manager in response to processing the markup language file.

First, Ferris does not disclose receiving text input to a GUI element enabled to receive text input displayed as a result of processing a markup language file comprising the description of that GUI element. In fact, Ferris expressly teaches away from receiving text to such GUI elements stating the following:

The FORM element has many limitations that the present invention addresses using individual applications or applets that run on a client. The applets are defined outside of a Web page and can be programmed to produce an unlimited number of input mechanisms to a user. (Ferris, col. 4, lines 47-53, emphasis added).

Accordingly, Ferris not only fails to disclose, but even teaches away from “markup language file comprising a description of a GUI, the description comprising a GUI element enabled to receive text input.”

Second, applicants submit that contrary to the assertion in the Office Action, the Applet Group Controller does not disclose a validation manager, as recited in claim 1. In particular, the limitation of “instantiating the validation manager in response to said processing the markup language file” that comprises a description of a GUI, is not disclosed. Ferris does not disclose the Applet Group Controller being instantiated as a result of processing of a markup language file that comprises the a description of a GUI. Indeed, as best as Ferris is understood, the reference suggests that the Applet Group Controller is instantiated independently, and not from tags in the markup language file processed by the browser. For example, Figure 5 of Ferris, and the corresponding text in the specification (Ferris, col. 12, line 55 - col. 13, line 48) disclose that the client creates components of Applet Group Controller in step 501 of Fig. 5, and only in step 503 (subsequent to step 501), the server transmits an HTML page (i.e. a markup language file) to the client. (Ferris, Fig. 5). Thus, the reference suggests that the Applet Group Controller is instantiated before the client computer receives the markup language file, and therefore, the Applet Group Controller cannot be instantiated as a result of processing of that markup language file. Accordingly, the description of Applet Group Controller in Ferris does not disclose the validation manager recited in claim 1.

Furthermore, the Examiner expressly acknowledged that Ferris fails to disclose: (1) “determining whether the text input received by the GUI element is valid text input”, and (2) “providing an indication that the text input received to the GUI element is invalid,” but stated that a combination with Strong would provide the missing elements, as to render claim 1 obvious. The Office Action indicated that a person of skill in the art would have found it obvious to combine the references as such combination would provide the capability for handling data validation in an information system. Applicants respectfully disagree.

First, applicants submit that the teachings of these two references cannot be combined as suggested in the Office Action. As noted, Ferris teaches away from using HTML forms, while these forms are required in Strong. For example, Ferris states that “the FORM element has many limitations that the present invention addresses using individual applications or applets that run on a client.” (Ferris, col. 4. lines 47-50). By contrast, Strong only discloses the use of forms. (See, Abstract, Summary of Invention at col. 2, lines 33-42, Detailed Description at col. 7, lines 7-10 stating “A method and apparatus for performing validation

and controlling processing of input data from electronic forms such as Hypertext Markup Language (HTML) forms is described.”). A person of skill in the art would have had no motivation to combine into one information processing system the teachings of two incompatible references. Accordingly, Ferris cannot be combined with Strong in the manner suggested in the Office Action.

Furthermore, even assuming that the teachings of Ferris and Strong could be combined, for the reasons set forth above, the combination would still fail to disclose a validation manager which is instantiated as a result of processing a markup language file, comprising description of GUI elements, as it is recited in the claims of the present application. Therefore, claim 1 is patentable over the prior art of record. The remaining claims 2-15 and 17-19 are dependent on claim 1, and therefore are also patentable.

In paragraph 7 of the Office Action, claim 16 was rejected was being unpatentable over Ferris in view of Strong as applied to claim 1 and further in view of the Microsoft Technology Overview. As was shown above, Ferris and Strong do not render claim 1 unpatentable either individually or in combination. The Microsoft Technology Overview reference discloses COM objects, but otherwise is not concerned with validation of user input, and therefore fails to supply the missing claim elements. Therefore, claim 16 is patentable over combination of Ferris, Strong, and Microsoft.

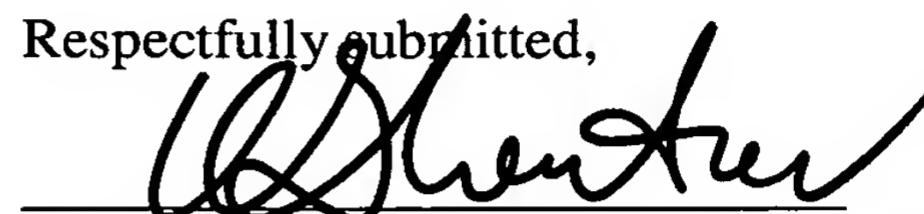
Based on the foregoing, all pending claims of this application, as amended, are believed to be patentable.

Conclusion

In light of the above amendments and remarks, the applicants respectfully request that the Examiner reconsider this application with a view towards allowance. The Examiner is invited to call the undersigned attorney if a telephone call could help resolve any remaining issues.

Date: February 23, 2006

Respectfully submitted,


Ognjan V. Shentov

38,051

(Reg. No.)

JONES DAY
222 East 41st Street
New York, New York 10017
(212) 326-3939